Table of Contents

Archiving Data

Archiving Data Overview

DRAFT

This article is being reviewed for completeness and technical accuracy.

The articles in this category provide basic information to assist you in archiving or retrieving files to/from the NAS long term storage systems, Lou1-2.

Mass Storage Systems Lou1-2 describes the disk and tape drives capacity, and how to find the storage system that is assigned to you.

Archiving Data in Compute Systems talks about backing up data from the home and /nobackup filesystems of Pleiades or Columbia to long term storage and what file transfer commands can be used for such tasks.

Quota Policy on Disk Space and Files provides information on the policy and soft and hard quota limits of disk space and files (inodes) of the home and /nobackup filesystems of Pleiades, Columbia and Lou.

Validating Files Transferred to Mass Storage by Size and Number describes a light way approach to validate the files transferred from Pleiades/Columbia to Lou.

Validating Files Transferred to Mass Storage with md5sum provides an example of checking the integrity (through the use of md5sum) of data transferred from Pleiades/Columbia to Lou.

Using the GNU Tar Utility provides examples of using *tar* to collect multiple files into a tar file for easier distribution or storage on Lou and to extract files out of a tar file later on if needed.

Using CPIO Utility describes the CPIO tool, which is similar to Tar, and provides a few examples on how to use it.

Data Migration Facility (DMF) Commands explains the usage of the DMF commands for managing files (such as listing, finding, migrating, un-migrating, or copying) stored on tapes.

Mass Storage Systems: Lou1 and Lou2

The NAS environment contains three mass storage systems, Lou1 and Lou2, to provide long-term data storage for users of our high-end computing systems. These storage systems are SGI Altix computers running the Linux operating system. The disk space for the three systems combined is about 290 terabytes (TB), which is split into filesystems ranging from 9-30 TB in size.

Which Lou System I Should Use?

Each user should be able to log into any of the Lou systems, but will only have storage space on the home filesystem of one of them. Follow the steps below to determine which system you should store data on.

1. Log in to either Lou1 or Lou2. For example:

   ```
   your_localhost% ssh nas_username@lou1.nas.nasa.gov
   ```

2. Type the command "mylou" to find out your mass storage host. For example:

   ```
   lou1% mylou

   Your Mass Storage host is lou2

   Store files there in your home directory, /u/your_nas_username
   ```

   Be aware that Lou1 and Lou2 do *not* share their home filesystems.

3. Use the home filesystem on Lou*X* (where *X* = 1 or 2) determined by the step above for your long-term storage. For example:

   ```
   pfe1% scp foo lou2:
   ```

Quota Limits On Lou

For Lou*X* (where *X* = 1 or 2) that is assigned to you, there are no disk quota limits on your home filesystem. On the other hand, there *are* limits on the number of files (inode):

- 250,000 inode soft limit (14-day grace period)
- 300,000 inode hard limit

See Policy on Disk Files Quotas for Lou for more information.

Data (Un)Migration Between Disk and Tapes

In addition to the disk space, Lou1 and 2 have a combined 64 LTO-4 tape drives. Each of the LTO-4 tapes holds 800 GB of uncompressed data. The total storage capacity is approximately 10 PB.

Data stored on Lou's home filesystems (disk) is automatically migrated to tapes whenever necessary to make space for more data. Two copies of your data are written to tape media in silos located in separate buildings.

Data migration (from disk to tape) and unmigration (from tape to disk) are managed by the SGI Data Migration Facility (DMF) and Tape Management Facility (TMF).

If you need some data that is only available on tapes, make sure to unmigrate the data from tape to your home filesystem on Lou before transferring it to other systems.

For more tips on how to use Lou more effectively, see Storage Best Practices.

Archiving Data in Compute Systems

DRAFT

This article is being reviewed for completeness and technical accuracy.

Archiving Data under Pleiades or Columbia Home Filesystems

Users' data under the home filesystem of any NAS HECC system (such as Pleiades or Columbia) are automatically backuped to Susan (an archive system accessible only to system administrators) under script control. The backup is done **daily**.

If you lost important data under a home filesystem and need to have it restored, please send a request to NAS Help Desk and provide the following:

- system name
- your username
- the directory and/or file name(s) that you wish to restore
- the date the data was last touched

Please note that disk space quota limits are imposed on the Pleiades and Columbia home filesystems. If you are over your disk space quota limits, reduce your usage by deleting unneeded files, copying important files to your home filesystem on Lou, or to storage space at your local site and then removing them from Pleiades or Columbia.

Archiving Data under Pleiades or Columbia Nobackup File Systems

Data under any /nobackup filesystem of any NAS HECC system (such as Pleiades /nobackupp[1,10-60], or Columbia /nobackup[21-24], /nobackup[1-2][a-j], etc.) are not backuped by NAS. Users are responsible to do their own backups.

Please note that disk space and inode quota limits are imposed on the Pleiades Lustre filesystems /nobackupp[1,10-60] and Columbia CXFS filesystems /nobackup1a-h and /nobackup2a-i. If you are over the soft quota limits, reduce the usage by removing some files and/or archive any important data to your home filesystem on Lou, or to the storage space at your local site.

In addition, when the overall usage of a /nobackup filesystem is near its capacity, files can be deleted by system administrators. Normally, the few top users are sent emails and asked to clean up their disk space.

See Policy on Disk Space Quotas of home and /nobackup File Systems for Pleiades/Columbia for more information.

What File Transfer Commands to Use

Note that the Columbia CXFS nobackup filesystems (e.g. /nobackup[1-2][a-i]) are mounted on the Mass Storage system (Lou1-2) that you are assigned to. As a result, you can log into LouX (where X is 1or 2) and copy the files from nobackup to your home directory on Lou. SGI has created a command called cxfscp, which is a tuned version of the cp command. You can copy files at up to 400MB/s sustained with cxfscp.

If you need to initiate the transfer from Columbia, then we recommend you use bbscp or bbftp if the file to be transferred does not need to be encrypted. If you need to encrypt the data, even within the HECC enclave, then scp should be used.

Transferring files from Pleiades to Mass Storage can be done with bbscp/bbftp or scp. Disk-to-disk copying to Mass Storage may be implemented in the near future.

Using the GNU Tar Utility

If you have multiple related files and directory trees that you would like to archive to Lou, it is better to collect them into a single archive file using the GNU tar utility. See Using the GNU Tar Utility to learn more.

Related Articles

- Pleiades Home Filesystem
- Columbia Home Filesystem
- Pleiades Lustre Filesystem
- Columbia CXFS Filesystems
- Local File Transfer Commands - cxfscp
- Remote File Transfer Commands

Quota Policy on Disk Space and Files

DRAFT

This article is being reviewed for completeness and technical accuracy.

Some NAS filesystems enforce quotas. Two kinds of quotas are supported: limits on the total disk space occupied by the user's files, and limits on how many files (represented by *inodes*) the user can store, irrespective of size. For quota purposes, directories count as files.

Further, there are two different limits: hard limits and soft limits. Hard limits cannot be exceeded, ever. Any attempt to use more than your hard limit will be refused with an error. Soft limits, on the other hand, can be exceeded temporarily. You can stay over your soft limit for a certain period of time (the grace period). If you remain over your soft limit for more than the grace period, the soft limit is enforced as a hard limit.

You will not be able to add or extend files until you get back under the soft limit. Usually, this means deleting unneeded files or copying important files elsewhere (perhaps the Lou archival storage system) and then removing them locally.

When you exceed your soft limit you will begin getting daily emails reminding you how long until the grace period expires. These are intended to be informative and not a demand to immediately remove files.

The following table summarizes the default space and inode quota limits enforced on Columbia, Pleiades and Lou.

Default Quotas on Disk Space and Files

	Columbia	**Pleiades**	**Lou**
$HOME	NFS	NFS	XFS
Space: soft	4 GB	8 GB	none
Space: hard	5 GB	10 GB	none
Inode: soft	none	none	250,000
Inode: hard	none	none	300,000
/nobackup	CXFS /nobackup1[a-g] /nobackup2[a-i]	Lustre /nobackupp[10-70]	N/A
Space: soft	200 GB	200 GB	N/A
Space: hard	400 GB	400 GB	N/A
Inode: soft	25,000	75,000	N/A

Inode: hard	50,000	100,000	N/A

Policy on Disk Space Quotas of home and /nobackup File Systems for Pleiades/Columbia

- If you exceed the soft quota, an email will be sent to inform you of your current disk space and how much of your grace period remains. It is expected that a user will exceed their soft limit as needed, however after 14 days, users who are still over their soft limit will have their batch queue access to Columbia/Pleiades disabled.
- If an account has been disabled for more than 14 days, then its Columbia/Pleiades data will be moved to the archive host, Lou, and kept there for 6 months before removal, unless the project lead requests to have the data moved to another account.
- If an account no longer has batch access to a system, then all data from that system should be moved off within 7 days (or sooner if the other projects need the space).
- If an account needs larger quota limits, please send an email justification to support@nas.nasa.gov. This will be reviewed by the HECC Deputy Project Manager, Bill Thigpen, for approval.

Policy on Disk File Quotas for Lou

- There is no quota for file space on Lou1or Lou2 because the data is written to tape. There is a quota on the number of files you can have. Currently there is a soft limit of 250,000 files and a hard limit of 300,000 files.
- There is a 14 day grace period if soft limit is exceeded. An email will be sent to inform you of your current disk space and how much of your grace period remains. It is expected that a user will exceed their soft limit as needed, however after 14 days, users who are still over their soft limit will be unable to archive files until they have reduced their use to below the soft limit.
- If an account needs larger quota limits, please send an email justification to support@nas.nasa.gov. This will be reviewed by the HECC Deputy Project Manager, Bill Thigpen, for approval.
- The maximum size of a file moved to Lou should not exceed 30% of the size of your home filesystem on Lou. If you need to move files larger than this, please contact the NAS Help Desk (support@nas.nasa.gov) for assistance.

Validating Files Transferred to Mass Storage by Size and Number

DRAFT

This article is being reviewed for completeness and technical accuracy.

It is a good practice to check if files are copied correctly to Mass Storage. The simplest and most light weight method to check files sent to Lou1or Lou2 is to compare the number of files and the disk space between the original and the copy.

Example:

```
pfe1% du -sk mydatadir
353760  mydatadir
pfe1% find mydatadir | wc -l
51
pfe1% scp -rp mydatadir lou2:

lou2% du -sk --apparent-size mydatadir
353684  mydatadir
lou2% find mydatadir | wc -l
51
```

Here the sizes are close and the number of files matches exactly.

Note that in most cases the directory size will not match exactly. The *--apparent-size* option is necessary on the Lou systems because the data may reside on tape, not disk.

Data Migration Facility DMF Commands

At the NAS facility, certain SGI systems (primarily the Lou filesystems) support a virtual storage manager feature called DMF (Data Migration Facility). Its purpose is to allow users to keep an increased volume of data in the files under their home directories by migrating those files that are not currently in use to offline backing storage, thereby allowing active disk space to be available for active files.

Migrated files are retrieved to active disk when you attempt to read or write to them, and the whole process is substantially transparent, aside from a possible time lag while a file is being retrieved.

The process of migrating files by backing them up from disk to another medium is implemented using STK tape silos.

DMF Commands

User commands associated with DMF are below, followed by usage examples:

dmls	Directory listing showing file migration status.
dmget	Retrieve migrated files to disk.
dmput	Cause files to migrate to backup on tape.
dmfind	Find files under a directory hierarchy.
dmcopy	Copy all or part of offline files.
dmattr	List attributes of files.

The *dmls* command

The *dmls* command is much like the standard *ls* command for file or directory listing, with the addition of an option for displaying the DMF status of files. Actually, *dmls* is derived from the GNU ls command, so it has a few extra "bells-and-whistles" compared to standard *ls* command; for details, see the **dmls man page**.

Example showing the extra status field:

```
% ls -l
total 64
-rw-r-----    1 aeneuman madmag     14713 Mar  1 17:02 file1
-rw-r-----    1 aeneuman madmag     17564 Mar  1 17:02 file2

% dmls -l
total 33
-rw-r-----   1 aeneuman madmag      14713 Mar  1 17:02 (REG) file1
-rw-r-----   1 aeneuman madmag      17564 Mar  1 17:02 (REG) file2
```

DMF has several possible states for files. The first three of them are the most likely to appear:

`REG`	File is a "regular" file that exists only online, on active disk.
`DUL`	File is "dual-state": identical copies of its data exist online and offline. The online copy will persist if there is no demand for free space in its filesystem. When free space is needed, the online copy of its data is released, leaving just the offline copy; the file becomes "offline." If you make any change to a dual-state file, the offline copy becomes out of date and invalid, and the file is once again a "regular" file.
`OFL`	File's directory entry remains but its data blocks are located offline only.
`MIG`	File is in the process of migrating offline.
`UNM`	File is in the process of un-migrating back online.
`NMG`	File is nonmigratable.
`INV`	File's DMF state is unknown or "invalid," usually because it is in a filesystem that does not use DMF.

The *dmget* command

The dmget command explicitly requests an offline file to be retrieved to disk. This command is not strictly required in order to retrieve offline files; any program that tries to use the file will cause it to be retrieved first.

The following example shows that referencing an offline file retrieves it after a pause. Notice that *C* was "offlined." Using the command *file C* caused this file to be retrieved and its status changed from *OFL* to *DUL*.

```
% dmls -l
total 1404
-rw-r-----   1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 11:20 (OFL) C

% file B
[immediate response:]
B:              English text

% file C
[pause while file is retrieved, then:]
C:              English text

% dmls -l
total 1404
-rw-r-----   1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 11:20 (DUL) C
```

You can retrieve the file explicitly with the command *dmget*. For example:

```
% dmls -l
total 220
-rw-r-----   1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 11:20 (OFL) C

% dmget C ; echo Done
[pause, then:]
Done
```

One motivation for retrieving files explicitly with *dmget* is that you may be able to save time. If you are working with several files that have been migrated at the same time and reside on the same offline tape cartridge, a *dmget* command that names all the files would be able to retrieve them all in a single mount of the tape.

By contrast, simply using the files, one after another, would cause the tape robot to fetch the tape cartridge, retrieve the first file and put away the tape, then go back and fetch the cartridge, retrieve the second file, put away the tape, and so on.

Another good reason for retrieving files explicitly with *dmget* is that you control when the retrieval takes place. For example, suppose you have a large, multiprocessor application that is going to read a currently migrated data file. You would prefer the retrieval process to take place with just your shell running, not when your main application has spawned several dozen processes on several dozen nodes, with all of them having to sit idle waiting for the file to be retrieved.

See the example under dmfind for an efficient method to recall all files in a directory and its subdirectories.

The *dmput* command

In filesystems that are under DMF automated space management, large files that have not been used for a while will be migrated offline automatically. The definitions of "large" and "a while" are established on each system by its system administrator.

You can invoke migration explicitly with the *dmput* command. Useful options are:

-r	Causes DMF to release a file's online disk space immediately, giving it Offline status. Otherwise, the file would be in Dual status until its disk space was needed for other files.
-w	Causes the *dmput* command to wait until migration has completed before returning control. Otherwise the *dmput* command returns immediately with the file in Migrating status.

Here's an example showing immediate return and changing state:

```
% dmls -l
total 1404
-rw-r-----   1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 12:43 (REG) C

% dmput C ; dmls -l
[immediately:]
total 1404
-rw-r-----   1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 12:43 (MIG) C

[then after a while:]
% dmls -l
total 1404
-rw-r-----   1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 12:43 (DUL) C
```

Example showing delayed return, and release of disk space:

```
% dmls -l
total 1404
-rw-r-----   1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 15:17 (REG) C

% dmput -r -w C ; dmls -l
[long pause:]
total 220
-rw-r-----   1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 15:17 (OFL) C
```

The *dmfind* command

The *dmfind* command is based on the GNU version of the find command and adds the ability to search for files in a given DMF state. This is handy for determining which files are offline and, therefore, candidates for retrieval with *dmget*.

Example:

```
% dmls -l
total 220
-rw-r-----   1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 15:17 (OFL) C

% dmfind  .  -state ofl  -print
```

```
./C

% dmget  `dmfind . -state ofl -print`
[pause:]

% dmls -l
total 1404
-rw-r-----   1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 15:17 (DUL) C
```

To efficiently recall all files in a directory named mydir and its subdirectories, use the following command:

```
% dmfind mydir -state mig -or -state ofl -print | dmget
```

The *dmcopy* command

The *dmcopy* command copies all or part of an offline file to a destination file, keeping the offline file offline.

When the offline file is being copied in its entirety, the only arguments needed are the two filenames.

Example:

```
% dmls -l
total 224
-rw-r-----   1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 15:17 (OFL) C

% dmcopy  C  newC
[pause:}

% dmls -l
total 1404
-rw-r-----   1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 15:17 (OFL) C
-rw-------   1 aeneuman madmag    1209300 Mar  4 15:06 (REG) newC
```

Dmcopy has options that allow copying just part of the offline file, and specifying offset locations in the source and destination files.

-l length	Specifies a data length to copy in bytes. The default is the whole size of the source file.
-o source-offset	Specifies a byte offset in the offline source file where reading is to begin. The default is zero; that is, reading begins at the start

	of the source file.
-d destination-offset	Specifies a byte offset in the destination file where writing is to begin. Any bytes in the destination file before this offset are zero filled. The default destination offset is whatever the source offset is.

Example: Skip two records of the offline source file and copy the next three records. Records are 2K bytes long.

```
% set RECDLEN=2048
% set NRECD=3
% @ LENGTH = $NRECD * $RECDLEN
% set NSKIP=2
% @ OFFSET = $NSKIP * $RECDLEN

% dmcopy  -l $LENGTH  -o $OFFSET  -d 0  C  newC
[pause:]

% dmls -l *C
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 15:17 (OFL) C
-rw-------   1 aeneuman madmag       6144 Mar  4 16:10 (REG) newC
```

Note: If the source file is a regular file, without an offline copy in DMF, the destination file is always zero length.

Note: The modes on the source file are ignored when creating the destination file. The modes on the destination file are 0666 (readable and writable by everyone), except where blocked by the current umask. Review your output files' permissions and reset them with *chmod* as needed.

The *dmattr* command

The *dmattr* command prints selected attributes of specified files. This is most useful in shell scripts.

Some options include:

-a attr1,attr2,...	Selects a subset of the reportable attributes.
-d delimiter	Specifies a one-character separator between adjacent values. A space is the default.
-l	Prints the attributes, one per line, with labels.

Examples:

```
% dmattr -l A
      bfid : 0
     emask : 0
```

```
   fhandle : 0100000000000188dede3a4b319c5b9000e000000000001000000000b3fd163
     flags : 0
     nregn : 0
     owner : 4771
      path : A
      size : 20155
     space : 20480
     state : REG

% dmattr A
0 0 0100000000000188dede3a4b319c5b9000e000000000001000000000b3fd163 0 0
4771 A 20155 20480 REG

% dmattr  -a owner,state  -d :  A
4771:REG

% foreach file ( * )
? if (`dmattr -a state $file` == OFL) then
? echo $file is offline
? endif
? end

C is offline
```